# Table of Contents

**Introduction to Splunk DFS Manager**.................................................................................................................................1  
   Splunk DFS Manager overview....................................................................................................................................................1  

**Get started with Splunk DFS Manager**.................................................................................................................................3  
   Install Splunk DFS Manager.......................................................................................................................................................3  
   Access Splunk DFS Manager.......................................................................................................................................................6  
   Overview homepage.................................................................................................................................................................8  
   Monitor dashboard.................................................................................................................................................................9  
   Upgrade Splunk DFS Manager.....................................................................................................................................................11  

**Using Splunk DFS Manager**..................................................................................................................................................12  
   Check the status of the compute cluster...........................................................................................................................12  
   Add DFS workers...............................................................................................................................................................12  
   Remove DFS workers.......................................................................................................................................................12  
   Change DFS settings.......................................................................................................................................................13  
   DFS Settings page...........................................................................................................................................................15  

**Troubleshoot Splunk DFS Manager**........................................................................................................................................17  
   Troubleshoot Splunk DFS Manager....................................................................................................................................17  
   Frequently asked questions (FAQs).........................................................................................................................................17  
   Splunk DFS Manager error messages....................................................................................................................................20  

**Release Notes**.......................................................................................................................................................................22  
   What's new in this release...................................................................................................................................................22  
   Known Issues.................................................................................................................................................................22  
   Fixed Issues...............................................................................................................................................................23  

**Additional resources**.............................................................................................................................................................24  
   Share data in Splunk DFS Manager.......................................................................................................................................24  
   Third-party software..........................................................................................................................................................26
Introduction to Splunk DFS Manager

Splunk DFS Manager overview

The Splunk DFS Manager app enables you to run Data Fabric Search (DFS). The app bundles, installs, and configures the compute cluster for your DFS deployment automatically. Additionally, the app helps you to continuously and seamlessly manage, configure, and monitor your compute cluster and customize the resource allocation within your DFS deployment from any search head through a scalable and adaptive user interface to run a DFS search.

Splunk only provides support for a compute cluster that is deployed using the Splunk DFS Manager app. If you install your compute cluster manually, Splunk isn't responsible for the support or maintenance of the compute cluster.

You can use the app irrespective of your deployment scenario and install the compute cluster on a standalone search head, standalone indexer, search head cluster, or an indexer cluster.

Splunk DFS Manager app is compatible with Python versions 2 and 3. For more information on the compatibility matrix for DFS and Splunk DFS manager app, see Compatibility matrix.

The Splunk DFS Manager app provides the following capabilities:

- High availability that allows an alternate search head captain to restart DFS master, in case a search head captain fails
- Add DFS workers on all or selected search peers
- Remove a DFS worker
- Restrict adding DFS workers to a particular site in a multi-site environment
- Monitor the health and resource usage of the compute cluster
- Enable compute cluster security

You must have administrator privileges to use the Splunk DFS Manager app to add or remove DFS workers or change the compute cluster settings.

Port security

The default Spark port is 8008 to launch DFS jobs and connect DFS workers. The default Spark Web UI port is 8009 to view the compute cluster. The default Spark history server port is 8010 to gather information on resource usage like CPU and memory allocation from event logs.

Allow only the search heads in the search head cluster and the search peers to access the compute cluster master and the Web UI port. You can block all other nodes from accessing the master and the Web UI port by using iptable and firewall rules.

Support

Contact Splunk Services if you run into issues trying to configure your compute cluster using the Splunk DFS Manager app.

Discuss the app for Splunk DFS Manager on Splunk Answers.

If you have a support contract, submit a case using the Splunk Support Portal.
For general Splunk platform support, see the Support and Services page.
Get started with Splunk DFS Manager

Install Splunk DFS Manager

Splunk DFS Manager app must be deployed on dedicated nodes of type indexers to run DFS workers. It is not recommended to use existing indexers that are currently ingesting data as DFS workers. Additional indexer nodes are required to run DFS workers.

- If you have an indexer cluster, you can add an additional dedicated indexer cluster to run DFS workers.
- If you are not using an indexer cluster, you can add additional dedicated indexers to run DFS workers.

Ensure that the app binaries are available on the DFS search heads and the dedicated indexers that you want to run as DFS workers. If two search heads in a non-clustered set up point to the same set of indexers, you must ensure that the app is installed on only one of the search heads. Failing to do so may cause inconsistencies in the app state.

Set all search heads to forward the data from the _internal indexes to the search peers to display the correct number of DFS searches and logs for all Spark instances. Otherwise, you do not see the logs for all the search heads in your DFS deployment. You can only see the logs for the local search head and the indexers.

For information on the compatibility matrix for DFS and Splunk DFS Manager, see Compatibility matrix.

Supported deployments

Use the following table to identify the supported deployment models for the Splunk DFS Manager app:

<table>
<thead>
<tr>
<th>Deployment</th>
<th>App installation method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Splunk instance running on an indexer</td>
<td>Not supported</td>
</tr>
<tr>
<td>Standalone deployment where search heads and indexers are all in one box</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Distributed non-clustered indexer deployment with one search head</td>
<td>Deploy the app on the search head and on every indexer provisioned to run as a DFS worker.</td>
</tr>
<tr>
<td>Distributed clustered indexer deployment without search head clustering</td>
<td>Deploy the app on the search head and every indexer of the additional indexer cluster provisioned to run as DFS workers.</td>
</tr>
<tr>
<td>Distributed clustered deployment with search head clustering</td>
<td>Deploy the app on every indexer of the additional indexer cluster provisioned to run as DFS workers and on all the search heads.</td>
</tr>
</tbody>
</table>

Using a dedicated Splunk indexer cluster is the recommended option for using Splunk DFS Manager to install the DFS compute cluster and connect it to the search head tier using the cluster master URI. The indexer cluster is dedicated to running DFS workers only and does not ingest any external data. Only the internal log files of the compute cluster will be indexed locally and may be searched, so that you can monitor and troubleshoot DFS seamlessly.
Splunk DFS Manager does not support the scenario where an indexer cluster that is running the app is part of a deployment with two search heads running as the Spark Master. The search heads running the Spark master may be in stand alone or search head cluster mode. This scenario is not supported because two search heads running as Spark masters and connecting to the same set of indexers (whether they are a part of an indexer cluster or standalone) could lead to a split brain. Therefore, the indexers may not be able to differentiate the current Spark master and may bounce between the two Spark masters resulting in a loss of service.

Splunk DFS Manager supports multiple search heads connected to the same set of indexers or distributed search peers but only one SH may run the DFS Manager application.

**Prerequisites to install Splunk DFS Manager**

Ensure that you meet the following prerequisites prior to installing the Splunk DFS Manager app:

- You have an existing Splunk deployment with at least one search head and one indexer.
- All indexers and search heads run on Splunk Enterprise version 8.0 or higher.
- Identify the number of DFS workers to run a DFS search.
- Configure the security settings on the DFS workers similar to the indexers (distributed search peers). For more information on configuring security settings, see Secure a DFS deployment.
- Ensure that ports 8008-8010 are open and available on each node to run the DFS master, DFS workers, and the history servers.

<table>
<thead>
<tr>
<th>Port name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFS compute cluster app web UI port</td>
<td>8005</td>
<td>Access application related information</td>
</tr>
<tr>
<td>Block manager port</td>
<td>8006</td>
<td>Block DFS manager on both the driver and the executor</td>
</tr>
<tr>
<td>Driver port</td>
<td>8007</td>
<td>Communications from the worker to the driver</td>
</tr>
<tr>
<td>Master/Worker port</td>
<td>8008</td>
<td>Communications between the master and the worker</td>
</tr>
<tr>
<td>Master/Worker web UI port</td>
<td>8009</td>
<td>Access master or worker related information</td>
</tr>
<tr>
<td>History server web UI port</td>
<td>8010</td>
<td>Access application historic data</td>
</tr>
</tbody>
</table>

For more ports about Spark ports, see Spark ports.

**Install Splunk DFS Manager**

1. Enable DFS on the standalone search head or the search head cluster by changing the `disabled` field to `false` in the `[dfs]` stanza of the `server.conf` configuration file.
2. Provision additional indexers to run as DFS workers.
   1. Provision new and dedicated indexers to run as DFS workers.
   2. Provision a new and dedicated indexer cluster.
3. Add the additional indexers as distributed search peers to the search head.
4. Download Splunk DFS Manager app from Splunkbase.
   1. Go to Splunk DFS Manager on Splunkbase.
   2. Click Download.
   3. Save the file to an accessible location.
5. Install the app on the search head and the search peers. For deployment specific app installation instructions, see Deployment specific app installation instructions.
6. Access the Splunk DFS Manager UI. For more information on accessing the UI of the Splunk DFS manager, see Access Splunk DFS Manager.
Deployment specific app installation instructions

Based on your deployment scenario, use the following instructions to install the app binaries in specific locations:

**Standalone search heads**

To install Splunk DFS Manager on a non-clustered search head, you can use the file system or Splunk Web.

**Using the file system**

1. Unzip the file.
2. Copy the directory to the correct location: On Linux: `$SPLUNK_HOME/etc/apps`

**Using Splunk Web**

1. On the Splunk bar, click the Applications menu, and select Manage Apps.
2. On the Apps page, click Install App from file.
3. Click Choose File, navigate to and select the package file for the Splunk DFS Manager, and then click Open.
4. Click Upload.

**Search head cluster**

You can install Splunk DFS Manager in a clustered environment by installing the app on each cluster member using the deployment server, the command line, the file system, or Splunk Web.

**Using the deployment server**

1. From the deployment server, install Splunk DFS Manager to the following location: `$SPLUNK_HOME/etc/shcluster/apps`.
2. Deploy the app bundle on all search heads.

Installing an app to the members of a search head cluster can cause a rolling restart.

**Using the command line**

At the command line (for Linux), enter the following: `$SPLUNK_HOME/bin/splunk install app $SPLUNK_HOME/etc/apps/

**Using the file system**

1. Unzip the file.
2. Copy the directory to the correct location: On Linux: `$SPLUNK_HOME/etc/apps`.

**Using Splunk Web**

1. On the Splunk bar, click the Applications menu, and select Manage Apps.
2. On the Apps page, click Install App from file.
3. Click Choose File, navigate to and select the package file for the Splunk DFS Manager, and then click Open.
4. Click Upload.
**Standalone indexers**

To install Splunk DFS Manager on a standalone indexer, download the app from Splunkbase to your local host, then install the app using Splunk Web.

1. Install Splunk DFS Manager to the following location: `$SPLUNK_HOME/etc/apps`
2. Restart Splunk Enterprise.

**Indexer clusters**

To use Splunk DFS Manager on the provisioned indexer cluster that runs as DFS workers, you must install the app onto the cluster master node. The cluster master pushes the app to all of the peer nodes.

1. On the master node, install Splunk DFS Manager to the following location: `$SPLUNK_HOME/etc/master-apps`
2. Distribute the app bundle to all the peer nodes.

Ensure that the app is installed in the `$SPLUNK_HOME/etc/slave-apps` directory if you are using an indexer cluster deployment. Otherwise, the Splunk DFS Manager app does not work as expected.

DFS master automatically starts on the search head in the standalone mode and on the search head captain in a search head cluster.

For more information on app deployment, see App deployment overview in the Splunk Enterprise Admin Manual.

For more information on deploying apps to clusters, both on search head cluster members and indexers, using the configuration bundle method, see Deploy apps to clusters in theSplunk Enterprise Admin Manual.

See the following documentation for details on setting up a search head cluster and an indexer cluster:

- Deploy a search head cluster in the Splunk Enterprise Distributed Search Manual.
- Update common peer configurations and apps in the Splunk Enterprise Managing Indexers and Clusters of Indexers Manual.
- Use the deployer to distribute apps and configuration updates in the Splunk Enterprise Distributed Search Manual.

**Access Splunk DFS Manager**

Access the Splunk DFS Manager app UI from the standalone search head or the search head captain in a clustered environment where the app is installed. Viewing the app UI from a standalone indexer is not currently supported.

In a search head clustering environment, the Splunk DFS Manager app is in a read-only view for all cluster members and you can configure the app only on the search head captain. For non search head clustering environments, the app should only be installed on a single search head, and that search head can be used for system configuration.
Upgrade to Safari 12 to use the Splunk DFS Manager UI on Safari. No limitations exist for Mozilla, Internet Explorer, or Firefox.

**Launch Splunk DFS Manager**

You can launch Splunk DFS Manager from the list of applications displayed on the Splunk Enterprise homepage. You can also launch Splunk DFS Manager from the Apps toolbar. Alternatively, you can launch Splunk DFS Manager by clicking **Apps > Manage Apps**.

Use the following steps to launch Splunk DFS Manager from the Search & Reporting app:

1. Navigate to the **Search & Reporting app**: `http://<searchhead_hostname>:8000/en-US/app/launcher/home`. If the app for Splunk DFS Manager is installed and enabled on your Splunk Enterprise deployment, it is listed as an app in the Apps toolbar.
2. Launch the app by clicking **Splunk DFS Manager**.

Use the following steps to launch Splunk DFS Manager from **Manage Apps**:

1. Click **Apps > Manage Apps**. A list of installed apps displays.
2. Find the app for Splunk DFS Manager in the list.
3. Click **Launch app** to access the user interface for the app.

The Splunk DFS Manager app consists of two main panels. The first panel **Overview** displays the app homepage. The second panel **Monitor** displays the monitoring dashboard.

**Identify component roles using Splunk DFS Manager**

Splunk DFS Manager behaves differently based on the component on which it is installed. Once installed, the app starts a modular input on the node and identifies its component role within the cluster.

Start Splunk DFS Manager to identify the following component roles in your deployment:

- Search head captain, which runs the compute cluster master
- Search head member, which might be assigned the role of a search head captain in case the search head captain fails
- Search peer, which works as an indexer and could be added as a DFS worker.

Depending on where Splunk DFS Manager is installed, the app loads in different ways:

- **Search head captain**: The app automatically boots up the compute cluster master on the search head captain.
- **Standalone search head**: The app automatically boots up the compute cluster master on the standalone search head.
- **Search peer**: The app waits to take action until you add the search peer as a worker to your deployment using the Splunk DFS Manager user interface. Once you add the search peer as a DFS worker to your deployment, a DFS worker instance boots up and is registered to the compute cluster master.

In case the search head captain fails, the role of the search captain may be assigned to another search head and the compute cluster master is stopped on the current captain and started on the new captain. If the search captain changes, all the DFS workers are notified of the new compute cluster master.
Overview homepage

The Splunk DFS Manager app Overview page consists of two sections. The first section consists of four tiles that provide information on the Splunk DFS Manager, DFS workers, as well as available vCPUs and memory in the compute cluster. The second section provides detailed information on the workers running on the compute cluster.

The following table describes the main elements in the top panel:

<table>
<thead>
<tr>
<th>Number</th>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DFS Manager</td>
<td>Information on compute cluster status, Web UI, and security.</td>
</tr>
<tr>
<td>2</td>
<td>Workers</td>
<td>Information on the number of enabled workers, sick workers, healthy workers, and available search peers.</td>
</tr>
<tr>
<td>3</td>
<td>vCPU</td>
<td>Information on vCPUs used by the workers and the total vCPUs available.</td>
</tr>
<tr>
<td>4</td>
<td>Memory</td>
<td>Information on used memory by the workers and the total memory available.</td>
</tr>
</tbody>
</table>

Use the following table for details on the DFS Manager tile in the top panel of the app homepage:

<table>
<thead>
<tr>
<th>UI component</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the compute cluster and whether it is in a Running state.</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>There is also an option to restart the compute cluster, if the compute cluster is not healthy.</td>
<td></td>
</tr>
<tr>
<td>Web UI</td>
<td>Link to the web UI of the compute cluster master</td>
<td>sh2:8009</td>
</tr>
<tr>
<td>Security</td>
<td>Information on compute cluster security</td>
<td>Enabled Disabled</td>
</tr>
</tbody>
</table>

Use the following table for details on the Workers tile in the top panel of the app homepage:

<table>
<thead>
<tr>
<th>UI component</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled Workers</td>
<td>Number of DFS workers that are enabled on the search peers</td>
<td>3</td>
</tr>
<tr>
<td>Workers Up</td>
<td>Number of DFS workers that are online on the search peers</td>
<td>4</td>
</tr>
<tr>
<td>Workers Sick</td>
<td>Number of DFS workers that are malfunctioning on the search peers</td>
<td>4</td>
</tr>
<tr>
<td>Workers Down</td>
<td>Number of DFS workers that are offline on the search peers</td>
<td>4</td>
</tr>
<tr>
<td>Total Search Peers</td>
<td>Total number of dedicated search peers that are registered to the Splunk DFS Manager app</td>
<td>4</td>
</tr>
</tbody>
</table>

Additionally, the app homepage also displays the following information:

- Number of vCPUs or cores used and available for the DFS workers
- Memory used and available for the DFS workers

The information on cores and memory is updated in real time and shows the resource utilization in the compute cluster as DFS jobs are run and completed.

Use the following table for details on the bottom panel of the app homepage that displays all the registered DFS workers in your DFS deployment and their resource utilization, including information on vCPUs, memory, and system usage:
The DFS workers will not be displayed unless they are added to your configuration. The default number of DFS workers displayed in the list is 20. Click on the drop-down list to customize the number of DFS workers (10, 20, 50, or 100) that you want to display on each page.

<table>
<thead>
<tr>
<th>UI component</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Peer UI</td>
<td>IP address and port number of the DFS worker</td>
<td>10.224.69.101:8089</td>
</tr>
<tr>
<td>Worker State</td>
<td>The state of the DFS worker</td>
<td>Up/Down</td>
</tr>
<tr>
<td>Cores or vCPUs Used/Total</td>
<td>Ratio of used and total cores for the DFS worker</td>
<td>2/6</td>
</tr>
<tr>
<td>Memory Used/Total</td>
<td>Ratio of used and total memory for the DFS worker</td>
<td>9.98 GB/97.7 GB</td>
</tr>
<tr>
<td>Max splunkd Usage</td>
<td>System utilization, which is the percentage of system compute power utilized</td>
<td>40%</td>
</tr>
<tr>
<td>Actions</td>
<td>Remove the DFS worker</td>
<td>Remove</td>
</tr>
</tbody>
</table>

The following screenshot displays an overview of the Splunk DFS Manager app homepage:

![Monitor dashboard](image)

**Monitor dashboard**

Use the Monitor dashboard to monitor the status of the Splunk DFS Manager app and resource utilization across a specified time range. You can also use the Monitor dashboard to inspect the application logs.

The Monitor dashboard consists of two panels. The first panel Resource utilization displays the vCPU and memory used and the search time to run DFS searches within a specified time range. The second panel Application status allows you to drill down at a component level within a specified time range and a specified log level for troubleshooting errors and warnings. A detailed view of the specified logs is displayed for troubleshooting.

**Resource utilization**

Use the resource utilization time range to select the time during which you want to monitor the resources used (vCPU and memory) to run DFS searches. Monitoring resources over the specified time range helps to plan the resources required to run the high volume DFS searches.

The following screenshot displays the Resource utilization panel on the Monitor dashboard for the DFS Manager app:
Application status

Specify the time range to narrow down the scope of the troubleshooting and identify the specific logs that must be inspected. You can also choose to inspect the logs of a specific search head or indexer. Additionally, you can also specify the log level (ERROR, WARNING, or INFO) that you want to inspect. This ability to customize the log display helps you to zoom in on specific log or component level errors and identify the root cause of the problem. You can drill down further into the logs for more granular inspection of specific events.

The following screenshot displays the Application status panel on the Monitor dashboard for the DFS Manager app:
Upgrade Splunk DFS Manager

Use the following instructions to upgrade the Splunk DFS Manager app based on your deployment model:

**Search head cluster**

1. Install the Splunk DFS Manager from Splunkbase.
2. Upload the app to `/etc/shcluster/apps` on the deployer.
3. Push the app to all of the peer nodes using the command line:
   
   CLI: `/opt/splunk/bin/splunk apply shcluster-bundle -target https://<search_head>:<management_port> -auth admin:Chang3d`

Do not upgrade Splunk DFS Manager from the app home page in the UI.

**Indexer cluster**

1. Install the Splunk DFS Manager from Splunkbase.
2. Upload the app to `/etc/master-apps` directory on the cluster master.
3. Push the app to all of the peer nodes using the command line or the UI.
   
   CLI: `/bin/splunk apply cluster-bundle --answer-yes -auth admin:<password>`
   
   UI: `http://<cluster_master>:<8000>/manager/system/clustering_push`

Do not upgrade Splunk DFS Manager from the app home page in the UI.

**Standalone node**

In the Search app UI list of apps, find Splunk DFS Manager app and click **Upgrade**.

Clear the browser cache of the browser you use to access Splunk Web to make sure that you access a fresh version of Splunk Web after upgrading. If you do not clear the browser cache, some pages might fail to load.
Using Splunk DFS Manager

Check the status of the compute cluster

Check the status of the compute cluster by performing the following tasks:

1. Launch the app for Splunk DFS Manager.
   This displays the homepage page of the Splunk DFS Manager app.
2. Check whether the compute cluster is in Running state by navigating to the Master tile in the top panel of the app homepage.

You can restart the compute cluster, if the compute cluster is not healthy. However, restarting the compute cluster stops all the jobs currently running.

For more detailed information on the Spark cluster displayed on the Splunk DFS Manager homepage, see Splunk DFS Manager app homepage.

Add DFS workers

You must have administrator privileges to use the Splunk DFS Manager app to add or remove DFS workers or change compute cluster settings. If you are using a search head cluster deployment, you must make these changes on the search head captain.

Add DFS workers to the Splunk Data Fabric Search (DFS) configuration by performing the following tasks:

1. Launch the app for Splunk DFS Manager to start the compute cluster automatically as a master, unless a specific role is identified.
   For more information on component roles, see Identify component roles.
2. Click Add Worker to launch the Add Workers dialog box.
3. In the Add Workers dialog box, select the search peers that you want to add as DFS workers.
   The search peers are listed by IP address, instance name, status of the search peer, system utilization, and site location that allows you to determine which nodes are better suited to be DFS workers.
4. Install DFS workers on all or a few of the Splunk Enterprise deployments by using the All or Selected filters.
   You can also select specific DFS workers by typing the IP address or similar information on the DFS worker on the filter bar.
5. Select search peers to add as workers if they don't index any data or if their Splunk platform usage is less than 20%.
6. Click Save to start Spark on the selected search peers.

You can expect to see increased resource usage as you add DFS workers to your deployment. For example: You may see 95% memory usage by dedicated DFS workers.

Remove DFS workers

You must have administrator privileges to use the Splunk DFS Manager app to add or remove DFS workers or change compute cluster settings. If you are using a search head cluster deployment, you must make these changes on the search head captain.
Remove DFS workers from the Splunk Data Fabric Search (DFS) configuration by performing the following tasks:

1. Launch the app for Splunk DFS Manager.
   This displays the overview page of the Splunk DFS Manager app.
2. To remove individual DFS workers, click **Remove** in the **Actions** column next to the DFS worker.
3. To remove multiple DFS workers, select the workers that you want to remove from your DFS deployment.
4. Click **Bulk Remove** to remove the selected DFS workers from your DFS deployment.

A minimum of one DFS worker is required to run any DFS job. Removing a DFS worker might impact the DFS jobs that are running.

**Change DFS settings**

You can use the **Change DFS Settings** page to change your compute cluster port settings, secure DFS, set site affinity, and define a DFS local directory to store DFS compute cluster logs and temporary data. The option to set site affinity is displayed in the UI only if you have opted to set a multisite cluster.

Any changes to the current compute cluster settings may require you to restart the compute cluster and may impact ongoing searches.

**Change port settings**

You can configure the ports for the DFS compute cluster using the Splunk DFS Manager app UI without editing any configuration files. Ensure the required ports are available, otherwise you may receive error messages when using the app. For more information on port configuration requirements, see Port configuration requirements.

You must have administrator privileges to use the Splunk DFS Manager app to configure ports. If you are using a search head cluster deployment, you must make these changes on the search head captain. Any changes to the current compute cluster settings may require you to restart the compute cluster and may impact ongoing searches.

Configure the ports for DFS compute cluster using the app UI by performing the following tasks:

1. Launch the app for Splunk DFS Manager.
2. Click on **Settings** to launch the **Change DFS Settings** dialog box.
3. In the **Change DFS Settings** dialog box, enter the port information.

<table>
<thead>
<tr>
<th>Port</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spark Port</td>
<td>8008</td>
</tr>
<tr>
<td>Spark Web UI Port</td>
<td>8009</td>
</tr>
<tr>
<td>Spark History Server Port</td>
<td>8010</td>
</tr>
</tbody>
</table>

4. Click **Save**.
   This restarts the DFS compute cluster.

**Change security settings**

You can enable DFS security using the Splunk DFS Manager app UI without editing any configuration files.

You must have administrator privileges to use the Splunk DFS Manager app to enable DFS security. If you are using a search head cluster deployment, you must make these changes on the search head captain. Any changes to the current
compute cluster settings may require you to restart the compute cluster and may impact ongoing searches.

Enable DFS security using the app UI by performing the following tasks:

1. Launch the app for Splunk DFS Manager.
2. Click on Settings to launch the Change DFS Settings dialog box.
3. In the Change DFS Settings dialog box, under Security.
4. Select the security setting by clicking on the Enable or Disable buttons.

Enabling security implies that the authentication handshake is done using a shared secret key and the data transfer between workers is encrypted. Disabling security implies encryption on data transmitted from indexers to workers is disabled.
5. Click Save to save the settings.

Configure the local directory

Configure a local directory for all nodes to store the logs for the DFS compute cluster and other temporary data. You may choose to use the default location at: $SPLUNK_HOME/var/log/spark. Alternatively, to manage space limitations, you can specify a local directory using the app UI to store the logs and data.

The app sets the configured log directory on each node. Any changes to the current compute cluster settings may require you to restart the compute cluster and may impact ongoing searches. You must be an administrator of the Splunk deployment to edit any settings, including setting the path to the local directory. Additionally, you must ensure that the user has read and write access to the local directory path.

Configure the local directory using the app UI by performing the following tasks:

1. Launch the Splunk DFS Manager app.
2. Click on Settings to launch the Change DFS Settings dialog box.
3. In the Change DFS Settings dialog box, under Local Directory, specify the path to the local directory to store the logs. For example: /opt/splunk/var/log/spark

   Specifying an absolute path and ensure that as the administrator you have read and write access privileges to this location. Also, ensure that the specified path exists and there is adequate space for the local directory.
4. Under Auto Cleanup, select Enable or Disable. Enabling auto cleanup deletes logs and data that are generated by the searches on all the DFS workers after a specified period of time, if a retention time is specified. Disabling auto cleanup may result in aborted searches if space becomes limited.
5. Under Retention Period, specify the time in minutes, hours or days for which you want to maintain the logs.
6. Click Save.

You can select DFS workers to process the search from a specific site to optimize bandwidth and costs. Set the site affinity for a DFS search using the Splunk DFS Manager app UI.

The option to set site affinity is displayed in the UI only if you have opted to set a multisite cluster. For more information on setting a multisite cluster, see Multisite deployment overview.

You must have administrator privileges to use the Splunk DFS Manager app to set site affinity. If you are using a search head cluster deployment, you must make these changes on the search head captain. Setting the site affinity may abort any ongoing DFS searches. Any changes to the current compute cluster settings may require you to restart the compute cluster and may impact ongoing searches.
Change the site affinity settings for DFS by performing the following tasks:

1. Launch the app for Splunk DFS Manager.
2. Click on Settings to launch the Change DFS Settings dialog box.
3. In the Change DFS Settings dialog box, under Site Affinity select the site from which you want to use the DFS workers to process the search.
4. Click Save.

DFS Settings page

Use the following table for details on the Change DFS Settings page:

<table>
<thead>
<tr>
<th>UI component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spark ports</td>
<td>Set the Spark port, Web UI port, and the history server port. For more information on changing port settings, see Change port settings.</td>
</tr>
<tr>
<td>Security</td>
<td>Toggle the security of the DFS compute cluster. For more information on changing security settings, see Change security settings. For information on port requirements, see Port configuration requirements. For information on securing a DFS deployment, see Secure a DFS deployment.</td>
</tr>
<tr>
<td>Local Directory</td>
<td>Set the DFS local directory path to store the DFS compute cluster logs and data. For more information on configuring the local directory, see Configure local directory.</td>
</tr>
<tr>
<td>Site Affinity</td>
<td>Set the site affinity to select DFS workers from a specific site. For more information on setting site affinity, see Set site affinity.</td>
</tr>
</tbody>
</table>

The following screenshot illustrates the Change DFS Settings page:
Please note that changing Spark Port, Security, or Worker Local Directory requires Spark to restart.

Spark Ports
Ensure that selected ports are open for access by Splunk DFS Manager.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Enable</th>
<th>Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spark Port</td>
<td>8882</td>
<td></td>
</tr>
<tr>
<td>Spark Web UI Port</td>
<td>8009</td>
<td></td>
</tr>
<tr>
<td>Spark History Server Port</td>
<td>8010</td>
<td></td>
</tr>
</tbody>
</table>

Security
Enabling DFS security may impact search performance due to encryption of data transfer.

Local Directory
Enter local directory path to store logs and data. The directory is created on all nodes.

Path: /opt/splunk/var/log/spark

Auto Cleanup
Temporary logs and data generated from the searches are maintained for the specified period of time.

Retention Period: 30 minutes
Troubleshoot Splunk DFS Manager

Perform the following tasks to troubleshoot the Splunk DFS Manager app:

1. Check the Overview page
2. Check the logs
3. Review the error messages

Check the Overview page

Use the Overview homepage of the Splunk DFS Manager to troubleshoot issues with the app. You can access the Overview page by launching the Splunk DFS Manager app. For more information on troubleshooting DFS, see Troubleshoot DFS. Ensure that your deployment meets all the Prerequisites to install the Splunk DFS Manager. Review the Common issues to verify if it is a frequently occurring issue.

Check the logs

Splunk DFS Manager app provides three kinds of logs:

- **Toast messages that provide simple feedback on common errors**
  You can access the app Overview page from the search head (or the search head captain) to view the common error messages.

- **Log session in the Monitor dashboard**
  Navigate to the Monitor dashboard by clicking on the Monitor panel and scrolling down to the Application Status monitor. The Application Status monitor helps you to analyze and troubleshoot installation issues while using the app.

- **Log files generated by the Splunk DFS Manager app and the DFS master and DFS workers**
  For more information on DFS log files, see DFS log files.

Review the error messages

For information on Splunk DFS Manager error messages, see Error messages.

Frequently asked questions (FAQs)

Review the Prerequisites to install the Splunk DFS Manager to ensure that you meet all the requirements to install the app. For more information on troubleshooting DFS, see Troubleshoot DFS.

Following are the common issues and frequently asked questions (FAQs) that you can use as a guide to troubleshoot the Splunk DFS Manager app:

Q. Why do I see a blank app page?
You may see a blank app page for the following reasons:

- You deployed the app with the indexer cluster.
- You tried to access the app on one of the indexers.

You must access the app UI from search head. If you continue to see a blank app page, please report this issue as a bug.

**Q. Why do I see a hanging UI even from search head?**

You may see a hanging UI for the following reasons:

- The version of the Splunk Enterprise deployment is not compatible. For more information on compatibility, see Compatibility matrix.
- The required ports are not open or available. Review the Prerequisites to install the Splunk DFS Manager. This issue is a known issue for app versions 1.2.0 and lower.

**Q. How can I change the port configurations if I do not want to use the default values?**

Changing the port configurations or any other settings in the `spark_app.conf` configuration file manually is not recommended. You may change the port configurations from the app UI from version 1.3.0 or higher.

If you do not want to open the default ports when using app versions prior to 1.3.0:

1. **Copy the `spark_app.conf` configuration file located at**
   - `$SPLUNK_HOME/etc/apps/splunk_dfs_manager/default/spark_app.conf` to the following location:
   - `$SPLUNK_HOME/etc/apps/splunk_dfs_manager/local/spark_app.conf`
2. **Next, modify the ports** `spark_master_port` and `spark_master_webui_port` and `spark_history_webui_port` in the `spark_app.conf` configuration file.

**Q. How do I verify whether the DFS master, DFS workers, or the history servers are running or not?**

You can review the status of the compute cluster from the app Overview page. If you see toast messages displaying errors or warnings, you may need to troubleshoot the issue.

If you find an issue, connect to the sick node using SSH. Run the following command `ps -ef | grep java` on your node and check to see whether the following processes are running or not.

**On all nodes:** `splunk_dfs_manager-1.2.0.jar`
**On all search heads:** `org.apache.spark.deploy.history.HistoryServer`
**On search head captain or on the only existing search head:** `org.apache.spark.deploy.master.Master`
**On indexers that are added as workers:** `org.apache.spark.deploy.worker`

Following is an example of the processes you may see on a search head when the DFS master is running:

```
splunk  3619  3354  0 Apr27 ?        00:02:20
/opt/splunk/bin/jars/vendors/java/OpenJDK8U-jre_x64_linux_hotspot_8u242b08/bin/java -jar
/opt/splunk/etc/apps/splunk_dfs_manager/jars/splunk_dfs_manager-1.2.0-alpha+b322.438fa38f3b6a.jar

splunk  3765  1  0 Apr27 ?        00:02:18
/opt/splunk/bin/jars/vendors/java/OpenJDK8U-jre_x64_linux_hotspot_8u242b08/bin/java -cp
/opt/splunk/etc/apps/splunk_dfs_manager/vendor/spark-2.3.3-bin-hadoop2.7/conf:/opt/splunk/etc/apps/splunk_dfs_manager/vendor/spark-2.3.3-bin-hadoop2.7/jars/* -Xmx1g org.apache.spark.deploy.master.Master --host sh3
```

18
Q. Why is the Spark master not running?

Check the related Spark process on the node and review the logs for the session on the dashboard.

You can troubleshoot why the Spark master is not running as follows:

- Ensure that your DFS deployment has at least one search head and one indexer, both running on different nodes. For more information on supported deployments, see Supported deployments.
- Check if the ports required by the Spark master are open and available.
- Check to see if the local directory path in the settings is writable by the Linux user who is running Splunk.

Q. Why is the Spark worker not running?

Check the related Spark process on the node and review the logs for the session on the dashboard.

You can troubleshoot why the Spark worker is not running as follows:

- Ensure that your DFS deployment has at least one search head and one indexer, both running on different nodes. For more information on supported deployments, see Supported deployments.
- Check if the two ports required by the Spark worker are open and available.
- Check to see if the local directory path in the settings is writable by the Linux user who is running Splunk.
- Check your network settings and ensure that the DFS worker node can communicate with the DFS master node using the following steps:
  1. Use the following command: `ps -ef | grep java` to list the worker process.
  2. Find the flag `--master spark://master-address:8008`.
  3. Ensure that you can access `master-address:8008`.

You may have some advanced network setting e.g. some proxy or port mapping, which may be blocking access.

Q. Why is the history server not running?

Check the related Spark process on the node and review the logs for the session on the dashboard.

You can troubleshoot why the history server is not running as follows:

- Check if the port required by the Spark history server is open and available.
- Check to see if the local directory path in the settings is writable by the Linux user who is running Splunk.

Q. Why am I not able to run a DFS search even when the compute cluster is up and running?

- Make sure that you enabled DFS on this search head. Enable DFSD by changing the `disabled` field to `false` in the `[dfs]` stanza of the `server.conf` configuration file.
- If you see the error `splunk_dfs_manager: Spark workers that are not resolvable and registered as indexers are not allowed when Splunk is managing Spark`, review your network settings.
Check the app **Overview** page to see if the IPs of the DFS workers are external IPs. The Splunk DFS manager app starts DFS workers with the internal IP by default for versions 1.2.0 and lower. Therefore, delete these search peers using **Settings -> Distributed search** and then re-add these search peers with their internal IPs. Then, run the DFS SPL search again. For more information on troubleshooting DFS, see Troubleshoot DFS.

**Q. Where can I view the logs for the Splunk DFS Manager app?**

The log file `<SPLUNK_HOME>/var/log/splunk/splunkd.log` contains the logs for the Splunk DFS Manager app. Additionally, use the following command: `grep Exec <SPLUNK_HOME>/var/log/splunk/splunkd.log | grep splunk_dfs_manager` to review the log entries only for this app. You can also view the logs from the **Monitor** dashboard from version 1.2.0 onwards of the app.

**Q. How can I enable the debug log?**

Navigate to **Settings -> Server settings -> Server logging**. Search for **ExecProcessor**, click on the first result, and set it to DEBUG. Debug log entries will be generated in the log file and you may view them in the dashboard log session shortly. This change will be re-set once Splunk is restarted.

**Q. Are there any other logs that I can use for troubleshooting?**

The logs for DFS master and DFS workers are stored in the local directory `$SPLUNK_HOME/var/log/spark` that you set by default and are also stored locally on each node. For example: You can connect to the DFS master using SSH and check the logs at: `<Local directory>/logs`.

**Splunk DFS Manager error messages**

You may encounter several error and warning messages when you work with the Splunk DFS Manager app to install and manage your compute cluster.

**Miscellaneous errors**

Use the following table to identify and troubleshoot miscellaneous error messages you might see when using the Splunk DFS Manager:

<table>
<thead>
<tr>
<th>Error message</th>
<th>Cause</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Management port error:</strong> Failed to access the management port.</td>
<td>The management port is mis-configured in the <code>web.conf</code> configuration file.</td>
<td>Set the <code>mgmtHostPort</code> in the <code>[settings]</code> stanza of the <code>web.conf</code> configuration file.</td>
</tr>
<tr>
<td><strong>Search head cluster error:</strong> Search head cluster has been detected, but no search head captain is present.</td>
<td>The search head cluster is mis-configured.</td>
<td>Configure your search head cluster. For more information on search head clustering, see Configure the search head cluster.</td>
</tr>
<tr>
<td><strong>Operating system error:</strong> DFS only runs on Linux x86-64 systems.</td>
<td>DFS runs only on Linux x86-64 architecture.</td>
<td>Deploy DFS on a Linux x86-64 architecture. For more information on platform compatibility, see Compatibility matrix.</td>
</tr>
<tr>
<td><strong>Configuration setting error:</strong> DFS is not enabled in the configuration file.</td>
<td>The configuration setting to enable DFS isn't set correctly.</td>
<td>Enable DFS by changing the <code>disabled</code> field to <code>false</code> in the <code>[dfs]</code> stanza of the <code>server.conf</code> configuration file.</td>
</tr>
<tr>
<td>Error message</td>
<td>Cause</td>
<td>Workaround</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Component identification error</td>
<td>The Splunk DFS Manager can't communicate with the REST API endpoints.</td>
<td>No workaround. Wait a few minutes for the search peers to be registered to the search head.</td>
</tr>
<tr>
<td>Spark configuration setting error</td>
<td>The DFS master can't communicate with the REST API endpoints.</td>
<td>Check the status of your Splunk Enterprise deployment. Verify that Splunk REST API service is active and the modular input for Splunk DFS Manager is running. For more information on Splunk REST API, see Splunk REST API.</td>
</tr>
<tr>
<td>Script error</td>
<td>Failure in the state machine thread.</td>
<td>Check $SPLUNK_HOME/var/log/splunk/splunkd.log for more information.</td>
</tr>
<tr>
<td>Splunk connection handler error</td>
<td>Failed to use the REST API due to network issues.</td>
<td>Check network connections and check $SPLUNK_HOME/var/log/splunk/splunkd.log for more information.</td>
</tr>
</tbody>
</table>

**Warning messages**

Use the following table to identify and troubleshoot the warning messages when using the app for Splunk DFS Manager:

<table>
<thead>
<tr>
<th>Warning message</th>
<th>Cause</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multisite warning</td>
<td>The Splunk DFS Manager can't communicate with the REST API endpoints.</td>
<td>No workaround. Wait a few minutes for a response.</td>
</tr>
<tr>
<td>Multisite cluster warning</td>
<td>Multi-site clustering is not set on the search head</td>
<td>Set multisite to true in the [clustermaster] stanza of the server.conf configuration file.</td>
</tr>
<tr>
<td>Multisite cluster warning</td>
<td>Site is not set on the search head for multi-site clustering deployment</td>
<td>Set the site for multisite clustering deployment. For more information, see Configure multisite cluster nodes.</td>
</tr>
<tr>
<td>Multi-site worker information</td>
<td>The multisite can't communicate with the REST API endpoints.</td>
<td>Check the status of your Splunk Enterprise deployment. Verify that Splunk REST API service is active. For more information on Splunk REST API, see Splunk REST API.</td>
</tr>
<tr>
<td>Missing peer error</td>
<td>Multiple causes: The search peer can't communicate with the REST API endpoints when a search peer is added as a worker. The search peer is down.</td>
<td>Check the status of the search peer.</td>
</tr>
<tr>
<td>Resource utilization information error</td>
<td>The Splunk DFS Manager can't communicate with the REST API endpoints.</td>
<td>No workaround. Wait a few minutes for a response.</td>
</tr>
</tbody>
</table>
Release Notes

What's new in this release

The following table highlights the new features in the Splunk DFS Manager release version 1.2.0:

<table>
<thead>
<tr>
<th>Number</th>
<th>Feature</th>
<th>Description</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Python compatibility</td>
<td>Splunk DFS Manager app supports Python versions 2 and 3.</td>
<td>For more information on Splunk DFS compatibility, see Compatibility matrix.</td>
</tr>
<tr>
<td>2</td>
<td>Ability to configure a local directory</td>
<td>Use the UI to configure a local directory that can store and cleanup logs generated from the searches.</td>
<td>For more information on using the Splunk DFS Manager app to configure a local directory, see Configure the local directory.</td>
</tr>
</tbody>
</table>

Known Issues

The following are known issues in the Splunk DFS Manager for this release:

<table>
<thead>
<tr>
<th>Date filed</th>
<th>Issue number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-03-24</td>
<td>PBS-526</td>
<td>spark history server showing &quot;Codec is not available&quot; messages repeatedly</td>
</tr>
<tr>
<td>2020-02-28</td>
<td>PBS-488, PBS-508</td>
<td>New releases of Splunk Enterprise will not be compatible with Splunk DFS Manager after updating JAVA version  WORKAROUND: For now, user need to manually change the JAVA_HOME in &quot;&lt;SPLUNK_HOME&gt;/&lt;SPLUNK_APP_FOLDER&gt;/splunk_dfs_manager/default/spark_app.conf&quot; to the new one, then restart Splunk to pick the change.</td>
</tr>
<tr>
<td>2020-02-19</td>
<td>PBS-471</td>
<td>Active workers remain connected to the Spark Master after the app is removed from the search peer. Result: Incorrect total vCPU, and memory is displayed Workaround: Restart Spark to spawn a new Spark master so that the total vCPU and memory displayed do not include the orphaned worker.</td>
</tr>
<tr>
<td>2020-02-06</td>
<td>PBS-442</td>
<td>Spark processes cannot start when the local directory is not configured correctly. Workaround: Check the dashboard and search for the error message &quot;Failed to start Spark process due to issues with this path&quot; in the logs. If the error occurs, change the path to indicate the correct local directory location.</td>
</tr>
<tr>
<td>2020-01-09</td>
<td>PBS-350</td>
<td>Set up data forwarding manually for all search heads to indexers. Workaround: Configure all search heads to forward data to indexers.</td>
</tr>
<tr>
<td>2019-10-09</td>
<td>PBS-243</td>
<td>Message &quot;Cannot connect to Spark master&quot; is displayed though Spark master has restarted. Workaround: Refresh the page to remove the error message &quot;Cannot connect to Spark master&quot; manually as Spark master has already restarted.</td>
</tr>
<tr>
<td>2019-10-03</td>
<td>PBS-216</td>
<td>DFS job fails with error for about one minute after a successful indexer restart due to a check for rogue workers. Workaround:</td>
</tr>
</tbody>
</table>
If using the standalone indexer setup with the Splunk DFS Manager app, wait for approximately a minute to run DFS searches after restarting an indexer with an active DFS worker even though the DFS worker is "healthy". You may receive the following error message: "Failed response from DFM. Error occurred while trying to start Data Fabric Coordinator. DFS workers that are not resolvable and registered as indexers are not allowed when Splunk is managing Spark." You can run a DFS search when this error no longer appears.

<table>
<thead>
<tr>
<th>Date filed</th>
<th>Issue number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-09-22</td>
<td>PBS-33</td>
<td>Connection issues between DFS workers and Spark master occur when Spark or Spark master Web UI is set to a port that is currently in use. Workaround: Verify that the ports that you want to use for Spark and Spark Web UI are available prior to using the Splunk DFS Manager app UI to set the ports.</td>
</tr>
</tbody>
</table>

**Fixed Issues**

The following are the fixed issues in the Splunk DFS Manager for this release.

<table>
<thead>
<tr>
<th>Date resolved</th>
<th>Issue number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-02-04</td>
<td>PBS-345</td>
<td>Only the search head captain reports that DFS is disabled when Splunk DFS Manager is installed. Other search heads in the cluster do not report that DFS is disabled.</td>
</tr>
</tbody>
</table>
## Additional resources

### Share data in Splunk DFS Manager

When Splunk DFS Manager is deployed on Splunk Enterprise, the Splunk platform sends anonymized usage data to Splunk Inc. ("Splunk") to help improve Splunk DFS Manager in future releases. For information about how to opt in or out and how the data is collected, stored, and governed, see Share data in Splunk Enterprise.

### What data is collected

Splunk DFS Manager collects the following basic usage information:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>app.session.splunkdfsmanager.process</strong></td>
<td>data.action=&quot;receive summary info&quot;</td>
<td>Identify system usage and average system sizes for possible pricing insights</td>
</tr>
</tbody>
</table>
| | | component: app.session.splunkdfsmanager.process  
| | | data: { [-]  
| | | action: receive summary info  
| | | app: splunk_dfs_manager  
| | | customInfo: { [-]  
| | | app: app-splunk-pbs  
| | | appVersion: 1.0.0  
| | | splunkBuild: 90d420c13150  
| | | splunkVersion: 8.0.0  
| | | }  
| | | page: overview  
| | | summary: { [-]  
| | | coresUsed: 0  
| | | enabledWorkers: 3  
| | | memoryUsed: 0  
| | | securityStatus: disabled  
| | | siteAffinity: all_sites  
| | | sparkMasterPort: 7777  
| | | sparkMasterWebUiPort: 8010  
| | | sparkStatus: running  
| | | totalCores: 48  
| | | totalMemory: 365562  
| | | totalSearchPeers: 3  
| | | workersDown: 0  
| | | workersSick: 0  
| | | workersUp: 3  
| | | }  
| | | deploymentID: 3a9a036a-9daa-59ee-bc3d-bf5f39870a11  
| | | eventID: 3696931e-6456-9a1c-a461-185ca763cf71  
| | | experienceID: 1c078320-0990-999b-750f-013eb41f79c0  
| | | optInRequired: 3  
| | | timestamp: 1570556029  
| | | userID: 6f03a59ea9d9fc03ba0917011dd5e4fe82b0afde4ae6c637ec9bd5bbba1905141  
| | | version: 3  
| | | visibility: anonymous,support  
| **app.session.splunkdfsmanager.process** | data.action="receive network message" | Identify common errors and the }  
| | | component: app.session.splunkdfsmanager.process  
| | | data: { [-]  
| | | page: overview  
| | | summary: { [-]  
| | | coresUsed: 0  
| | | enabledWorkers: 3  
| | | memoryUsed: 0  
| | | securityStatus: disabled  
| | | siteAffinity: all_sites  
| | | sparkMasterPort: 7777  
| | | sparkMasterWebUiPort: 8010  
| | | sparkStatus: running  
| | | totalCores: 48  
| | | totalMemory: 365562  
| | | totalSearchPeers: 3  
| | | workersDown: 0  
| | | workersSick: 0  
| | | workersUp: 3  
| | | }  
| | | deploymentID: 3a9a036a-9daa-59ee-bc3d-bf5f39870a11  
| | | eventID: 3696931e-6456-9a1c-a461-185ca763cf71  
| | | experienceID: 1c078320-0990-999b-750f-013eb41f79c0  
| | | optInRequired: 3  
| | | timestamp: 1570556029  
| | | userID: 6f03a59ea9d9fc03ba0917011dd5e4fe82b0afde4ae6c637ec9bd5bbba1905141  
| | | version: 3  
| | | visibility: anonymous,support  

24
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| frequency of the warnings and / or errors | action: receive network message  
app: splunk_dfs_manager  
customInfo: ( [-]  
app: app-splunk-pbs  
appVersion: 1.0.0  
splunkBuild: 0c4e05e2810b8c9d059239e2421042a0e251b1c6  
splunkVersion: 20190916  
)  
endpoint: /sparkclusterinfo?add_worker=True&filter_detained=true  
messageType: spark_cluster_info_error  
page: overview  
type: error  
| component: app.session.splunkdfsmanager.interact  
data: ( [-]  
action: track filter type  
app: splunk_dfs_manager  
customInfo: ( [-]  
app: app-splunk-pbs  
appVersion: 1.0.0  
splunkBuild: 0c4e05e2810b8c9d059239e2421042a0e251b1c6  
splunkVersion: 20190916  
)  
filterType: text  
page: overview  
| component: app.session.splunkdfsmanager.interact  
data: ( [-]  
action: receive settings  
app: splunk_dfs_manager  
customInfo: ( [-]  
app: app-splunk-pbs  
appVersion: 1.0.0  
splunkBuild: 0c4e05e2810b8c9d059239e2421042a0e251b1c6  
splunkVersion: 20190916  
)  
page: overview  

---

Track filtering activities on filterable fields that are not supported by Splunk DFS Manager supports filtering using the <Search Peer URI> and <Worker State> fields.

Identify the frequency with which Spark settings are changed.
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Splunk DFS Manager contains some libraries that were written by others and are being redistributed as part of the app under their respective open source licenses. We want to thank the contributors of these projects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apache Spark</td>
<td>Version 2.3.3</td>
<td><a href="http://spark.apache.org/">http://spark.apache.org/</a></td>
</tr>
<tr>
<td>Apache License</td>
<td>Version 2.0, January 2004</td>
<td><a href="http://www.apache.org/licenses/">http://www.apache.org/licenses/</a></td>
</tr>
</tbody>
</table>

**TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION**

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and You must cause any modified files to carry prominent notices stating that You changed the files; and You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole,
provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

ConfigParser

Version: 3.7

https://github.com/python/cpython/blob/3.7/Lib/configparser.py

This module provides the ConfigParser class which implements a basic configuration language which provides a structure similar to what's found in Microsoft Windows INI files. You can use this to write Python programs which can be customized by end users easily.

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.
In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td></td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible,

because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011,
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark
sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

json

Version 3.7

https://github.com/python/cpython/blob/3.7/Lib/json/__init__.py

json exposes an API familiar to users of the standard library marshal and pickle modules.

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible?</th>
</tr>
</thead>
</table>

32
<table>
<thead>
<tr>
<th>Version</th>
<th>License</th>
<th>Year</th>
<th>Organization</th>
<th>GPL-compatible</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>CNRI</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible,

because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.


3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF shall not be liable to Licensee or any other users of Python for any incidental, special, or consequential damages or loss as a result of modifying, distributing, or otherwise using Python, or any derivative thereof, even if advised of the possibility thereof.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.
1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia’s conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT
Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

**OS**

Version 3.7

https://github.com/python/cpython/blob/3.7/Lib/os.py

This module provides a portable way of using operating system dependent functionality

### A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python’s principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnr resta on.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td></td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>
Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.


3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All
Rights Reserved” are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI’s License Agreement, Licensee may substitute the following text (omitting the quotes): “Python 1.6.1 is made available subject to the terms and conditions in CNRI’s License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013”.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia’s conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

socket

Version 3.7

https://github.com/python/cpython/blob/3.7/Lib/socket.py

This module provides access to the BSD socket interface.

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td></td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible,
because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.


3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or
binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.
4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

urlib

Version 3.7

https://github.com/python/cpython/tree/3.7/Lib/urlib/

urllib3 is a powerful, sanity-friendly HTTP client for Python.
MIT License

Copyright (c) 2008-2019 Andrey Petrov and contributors (see CONTRIBUTORS.txt)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to
whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.